Fast algorithms for ℓ-adic towers over finite fields

Luca De Feo
Laboratoire PRiSM
Université de Versailles
luca.de-feo@uvsq.fr

Javad Doliskani
Computer Science
Department
Western University
jdoliska@uwo.ca

Éric Schost
Computer Science
Department
Western University
eschost@uwo.ca

ABSTRACT

Inspired by previous work of Shoup, Lenstra-De Smit and Couveignes-Lercier, we give fast algorithms to compute in (the first levels of) the ℓ-adic closure of a finite field. In many cases, our algorithms have quasi-linear complexity.

Categories and Subject Descriptors

F.2.1 [**Theory of computation**]: Analysis of algorithms and problem complexity—*Computations in finite fields*; G.4 [**Mathematics of computing**]: Mathematical software

General Terms

Algorithms,Theory

Keywords

Finite fields, irreducible polynomials, extension towers, algebraic tori, Pell's equation, elliptic curves.

1. INTRODUCTION

Building arbitrary finite extensions of finite fields is a fundamental task in any computer algebra system. For this, an especially powerful system is the "compatibly embedded finite fields" implemented in Magma [2, 3], capable of building extensions of any finite field and keeping track of the embeddings between the fields.

The system described in [3] uses linear algebra to describe the embeddings of finite fields. From a complexity point of view, this is far from optimal: one may hope to compute and apply the morphisms in quasi-linear time in the degree of the extension, but this is usually out of reach of linear algebra techniques. Even worse, the quadratic memory requirements make the system unsuitable for embeddings of large degree extensions. Although the Magma core has evolved since the publication of the paper, experiments in Section 5 show that embeddings of large extension fields are still out of reach.

In this paper, we discuss an approach based on polynomial arithmetic, rather than linear algebra, with much better performance. We consider here one aspect of the question, ℓ-adic towers; we expect that this will play an important role towards a complete solution.

Let q be a power of a prime p, let $\mathbb{F}_q$ be the finite field with q elements and let ℓ be a prime. Our main interest in this paper is on the algorithmic aspects of the *ℓ-adic closure* of $\mathbb{F}_q$, which is defined as follows. Fix arbitrary embeddings

$$\mathbb{F}_q \subset \mathbb{F}_{q^\ell} \subset \mathbb{F}_{q^{\ell^2}} \subset \cdots ;$$

then, the ℓ-adic closure of $\mathbb{F}_q$ is the infinite field defined as

$$\mathbb{F}_q^{(\ell)} = \bigcup_{i\geq 0} \mathbb{F}_{q^{\ell^i}}.$$

We also call an *ℓ-adic tower* the sequence of extensions $\mathbb{F}_q, \mathbb{F}_{q^\ell}, \dots$ In particular, they allow us to build the algebraic closure $\bar{\mathbb{F}}_q$ of $\mathbb{F}_q$, as there is an isomorphism

$$\bar{\mathbb{F}}_q \cong \bigotimes_{\ell \text{ prime}} \mathbb{F}_q^{(\ell)}, \tag{1}$$

where the tensor products are over $\mathbb{F}_q$; we will briefly mention below the algorithmic counterpart of this equality.

We present here algorithms that allow us to "compute" in the first levels of ℓ-adic towers (in a sense defined hereafter); at level i, our goal is to be able to perform all basic operations in quasi-linear time in the extension degree ℓ^i. We do not discuss the representation of the base field $\mathbb{F}_q$, and we count operations $\{+, -, \times, \div\}$ in $\mathbb{F}_q$ at unit cost.

The techniques we use are inspired by those in [6], which dealt with the Artin-Schreier case $\ell = p$ (see also [7], which reused these ideas in the case $\ell = 2$): we construct families of irreducible polynomials with special properties, then give algorithms that exploit the special form of those polynomials to apply the embeddings. Because they are treated in the references [6, 7], *we exclude the cases $\ell = p$ and $\ell = 2$.*

The field $\mathbb{F}_{q^{\ell^i}}$ will be represented as $\mathbb{F}_q[X_i]/\langle Q_i\rangle$, for some irreducible polynomial $Q_i \in \mathbb{F}_q[X_i]$. Letting x_i be the residue class of X_i modulo Q_i endows $\mathbb{F}_{q^{\ell^i}}$ with the monomial basis

$$\mathbf{U}_i = (1, x_i, x_i^2, \dots, x_i^{\ell^i-1}). \tag{2}$$

Let $\mathsf{M} : \mathbb{N} \to \mathbb{N}$ be such that polynomials in $\mathbb{F}_q[X]$ of degree less than n can be multiplied in $\mathsf{M}(n)$ operations in $\mathbb{F}_q$, under the assumptions of [30, Ch. 8.3]; using FFT multiplication, one can take $\mathsf{M}(n) \in O(n \log(n) \log\log(n))$. Then, multiplications and inversions in $\mathbb{F}_q[X_i]/\langle Q_i\rangle$ can be done in respectively $O(\mathsf{M}(\ell^i))$ and $O(\mathsf{M}(\ell^i)\log(\ell^i))$ operations in $\mathbb{F}_q$ [30, Ch. 9-11]. This is almost optimal, as both results are quasi-linear in $[\mathbb{F}_{q^{\ell^i}} : \mathbb{F}_q] = \ell^i$.

Condition	Initialization	Q_i, T_i	Lift, push
$q = 1 \bmod \ell$	$O_e(\log(q))$	$O(\ell^i)$	$O(\ell^i)$
$q = -1 \bmod \ell$	$O_e(\log(q))$	$O(\ell^i)$	$O(\mathsf{M}(\ell^i)\log(\ell^i))$
$-$	$O_e(\ell^2 + \mathsf{M}(\ell)\log(q))$	$O(\mathsf{M}(\ell^{i+1})\mathsf{M}(\ell)\log(\ell^i)^2)$	$O(\mathsf{M}(\ell^{i+1})\mathsf{M}(\ell)\log(\ell^i))$
$4\ell \le q^{1/4}$	$O_{\tilde{e}}(\ell\log^5(q) + \ell^3)$ (bit)	$O_e(\ell^2 + \mathsf{M}(\ell)\log(\ell q) + \mathsf{M}(\ell^i)\log(\ell^i))$	$O(\mathsf{M}(\ell^i)\log(\ell^i))$
$4\ell \le q^{1/4}$	$O_{\tilde{e}}(\ell\log^5(q))$ (bit) $+ O_e(\mathsf{M}(\ell)\sqrt{q}\log(q))$	$O_e(\log(q) + \mathsf{M}(\ell^i)\log(\ell^i))$	$O(\mathsf{M}(\ell^i)\log(\ell^i))$

Table 1: Summary of results

Computing embeddings requires more work. For this problem, it is enough consider a pair of consecutive levels in the tower, as any other embedding can be done by applying repeatedly this elementary operation. Following again [6], we introduce two slightly more general operations, *lift* and *push*.

To motivate them, remark that for $i \ge 2$, $\mathbb{F}_{q^{\ell^i}}$ has two natural bases as a vector space over $\mathbb{F}_q$. The first one is via the monomial basis $\mathbf{U}_i$ seen above, corresponding to the univariate model $\mathbb{F}_q[X_i]/\langle Q_i\rangle$. The second one amounts to seeing $\mathbb{F}_{q^{\ell^i}}$ as a degree ℓ extension of $\mathbb{F}_{q^{\ell^{i-1}}}$, that is, as

$$\mathbb{F}_q[X_{i-1}, X_i]/\langle Q_{i-1}(X_{i-1}), T_i(X_{i-1}, X_i)\rangle, \qquad (3)$$

for some polynomial T_i monic of degree ℓ in X_i, and of degree less than ℓ^{i-1} in X_{i-1}. The corresponding basis is bivariate and involves x_{i-1} and x_i:

$$\mathbf{B}_i = (1, \dots, x_{i-1}^{\ell^{i-1}-1}, \dots, x_i^{\ell-1}, \dots, x_{i-1}^{\ell^{i-1}-1}x_i^{\ell-1}). \qquad (4)$$

Lifting corresponds to the change of basis from $\mathbf{B}_i$ to $\mathbf{U}_i$; *pushing* is the inverse transformation.

Lift and push allow us to perform embeddings as a particular case, but they are also the key to many further operations. We do not give details here, but we refer the reader to [6, 7, 16] for examples such as the computation of relative traces, norms or characteristic polynomials, and applications to solving Artin-Schreier or quadratic equations, given in [6] and [7] for respectively $\ell = p$ and $\ell = 2$.

Table 1 summarizes our main results. Under various assumptions, it gives costs (counted in terms of operations in $\mathbb{F}_q$) for initializing the construction, building the polynomials Q_i and T_i from Eq.(3), and performing lift and push. $O_e(\ )$ indicates probabilistic algorithms with expected running time, and $O_{\tilde{e}}(\ )$ indicates the additional omission of logarithmic factors. Two entries mention bit complexity, as they use an elliptic curve point counting algorithm.

In all cases, our results are close to being linear-time in ℓ^i, up to sometimes the loss of a factor polynomial in ℓ. Except for the (very simple) case where $q = 1 \bmod \ell$, these results are new, to the best of our knowledge. To otbain them, we use two constructions: the first one (Section 2) uses cyclotomy and descent algorithms; the second one (Section 3) relies on the construction of a sequence of fibers of isogenies between algebraic groups.

These constructions are inspired by previous work due to respectively Shoup [25, 26] and Lenstra / De Smit [19], and Couveignes / Lercier [4]. We briefly discuss them here and give more details in the further sections.

Lenstra and De Smit [19] address a question similar to ours, the construction of the ℓ-adic closure of $\mathbb{F}_q$ (and of its algebraic closure), with the purpose of standardizing it. The resulting algorithms run in polynomial time, but (implicitly) rely on linear algebra and multiplication tables, so quasi-linear time is not directly reachable. References [25, 26, 4] discuss a related problem, the construction of irreducible polynomials over $\mathbb{F}_q$; the question of computing embeddings is not considered. Note that the results in [4] are *quasi-linear*; they rely however on an algorithm by Kedlaya and Umans [13] that works only in a boolean model, and as a result share this specificity.

To conclude the introduction, let us mention a few applications of our results. A variety of computations in number theory and algebraic geometry require constructing new extension fields and moving elements from one to the other. As it turns out, in many cases, the ℓ-adic constructions considered here are sufficient: two examples are [5, 9], both in relation to torsion subgroups of Jacobians of curves.

The main question remains of course the cost of computing in arbitrary extensions. As showed by Eq. (1), this boils down to the study of ℓ-adic towers, as done in this paper, together with algorithms for computing in *composita*. References [25, 26, 4] deal with related questions for the problem of computing irreducible polynomials; a natural follow-up to the present work is to study the cost of embeddings and similar changes of bases in this more general context.

2. QUASI-CYCLOTOMIC TOWERS

In this section, we discuss a construction of the ℓ-adic tower over $\mathbb{F}_q$ inspired by previous work of Shoup [25, 26], Lenstra-De Smit [19] and Couveignes-Lercier [4]. The results of this section establish rows 1 and 3 of Table 1.

The construction starts by building an extension $\mathbb{K}_0 = \mathbb{F}_q[Y_0]/\langle P_0\rangle$, such that the residue class y_0 of Y_0 is a non ℓ-adic residue in $\mathbb{K}_0$ (we discuss this in more detail in the first subsection); we let r be the degree of P_0.

By [15, Th. VI.9.1], for $i \ge 1$, the polynomial $Y_i^{\ell^i} - y_0 \in \mathbb{K}_0[Y_i]$ is irreducible, so that $\mathbb{K}_i = \mathbb{K}_0[Y_i]/\langle Y_i^{\ell^i} - y_0\rangle$ is a field with $q^{r\ell^i}$ elements. If we let y_i be the residue class of Y_i in $\mathbb{K}_i$, these fields are naturally embedded in one another by the isomorphism $\mathbb{K}_{i+1} \simeq \mathbb{K}_i[Y_{i+1}]/\langle Y_{i+1}^{\ell} - y_i\rangle$; in particular, the relation $y_{i+1}^{\ell} = y_i$ holds.

In order to build $\mathbb{F}_{q^{\ell^i}}$, we apply a descent process, for which we follow an idea of Shoup's. For $i \ge 0$, let x_i be the trace of y_i over a subfield of index r:

$$x_i = \sum_{j=0}^{r-1} y_i^{q^{\ell^i j}}. \qquad (5)$$

Then, [25, Th. 2.1] proves that $\mathbb{F}_q(x_i) = \mathbb{F}_{q^{\ell^i}}$ (see Figure 1). In particular, the minimal polynomials of $x_1, x_2, \dots$ over $\mathbb{F}_q$ are the irreducible polynomials Q_i we are interested in.

We will show here how to compute these polynomials, the polynomials T_i introduced in Eq. (3) and how to perform lift and push. To this effect, we will define more general minimal polynomials: for $0 \le j \le i$, we will let $Q_{i,j} \in \mathbb{F}_q(x_j)[X_i]$ be the minimal polynomial of x_i over $\mathbb{F}_q(x_j)$, so

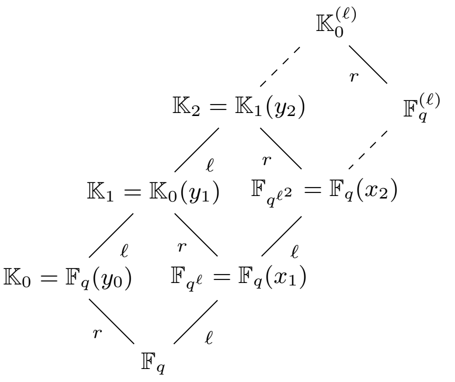
$\mathbb{K}_0^{(\ell)}$
r
$\mathbb{K}_2 = \mathbb{K}_1(y_2)$
$\mathbb{F}_q^{(\ell)}$
ℓ
r
$\mathbb{K}_1 = \mathbb{K}_0(y_1)$
$\mathbb{F}_{q^{\ell^2}} = \mathbb{F}_q(x_2)$
ℓ
r
ℓ
$\mathbb{K}_0 = \mathbb{F}_q(y_0)$
$\mathbb{F}_{q^\ell} = \mathbb{F}_q(x_1)$
r
ℓ
$\mathbb{F}_q$

Figure 1: The ℓ-adic towers over $\mathbb{F}_q$ and $\mathbb{K}_0$.

that $Q_{i,j}$ has degree ℓ^{i-j}, with in particular $Q_{i,0} = Q_i$ and $Q_{i,i-1} = T_i(x_{i-1}, X_i)$.

In Subsections 2.2 and 2.3, we discuss favorable cases, where ℓ divides respectively $q-1$ and $q+1$. The first case is folklore; it yields the fastest and simplest algorithms; our results for the second case are close to, but distinct from, previous work of Gurak [10] – we will revisit these cases in Section 3 and account for their naming convention. Our results in the general case (Subsection 2.4) are slower, but still quasi-linear in ℓ^i, up to a factor polynomial in ℓ.

Shoup used this setup to compute Q_i in time quadratic in ℓ^i [26, Th. 11]. It is noted there that using *modular composition* techniques [30, Ch. 12], this behavior could be improved to get a subquadratic exponent in ℓ^i, up to an extra cost polynomial in ℓ. For $\ell = 3$ (where we are in one the first two cases), Couveignes and Lercier make a similar remark in [4, § 2.4]; using a result by Kedlaya and Umans [13] for modular composition, they derive for any $\varepsilon > 0$ a cost of $3^{i(1+\varepsilon)}O(\log(q))$ *bit* operations, up to polynomial terms in $\log\log(q)$.

In this section, and in the rest of this paper, if L/K is a field extension, we write $\mathrm{Tr}_{L/K}$, $\mathrm{N}_{L/K}$ and $\mathrm{Gal}_{L/K}$ for the trace, norm and Galois group of the extension. Recall also that the notation $O_e(\ )$ indicates an *expected* running time.

2.1 Finding P_0

To determine P_0, we compute the ℓ-th cyclotomic polynomial $\Phi_\ell \in \mathbb{Z}[X_0]$ and factor it over $\mathbb{F}_q[X_0]$: by [26, Th. 9], this takes $O_e(\mathsf{M}(\ell)\log(\ell q))$ operations in $\mathbb{F}_q$.

Over $\mathbb{F}_q[X_0]$, Φ_ℓ splits into irreducible factors of the same degree r, where r is the order of q in $\mathbb{Z}/\ell\mathbb{Z}$ (so r divides $\ell-1$); let F_0 be one of these factors. By construction, there exist non ℓ-adic residues in $\mathbb{F}_q[X_0]/\langle F_0\rangle$. Once such a non-residue y_0 is found, we simply let P_0 be its minimal polynomial over $\mathbb{F}_q$ (which still has degree r); given y_0, computing P_0 takes $O(r^2)$ operations in $\mathbb{F}_q$.

Following [25, 26, 4], we pick y_0 at random: we expect to find a non-residue after $O(1)$ trials; by [26, Lemma 15], each takes $O_e(\mathsf{M}(\ell)\log(r) + \mathsf{M}(r)\log(\ell)\log(r) + \mathsf{M}(r)\log(q))$ operations in $\mathbb{F}_q$. An alternative due to Lenstra and De Smit is to take iterated ℓ-th roots of $X_0 \bmod F_0$ until we find a non-residue: this idea is helpful in making the construction canonical, but more costly, so we do not consider it.

2.2 T_1-type extensions

We consider here the simplest case, where ℓ divides $q-1$; the (classical) facts below give the first row of Table 1.

In this case, Φ_ℓ splits into linear factors over $\mathbb{F}_q$ (so $r = 1$). The polynomial P_0 is of the form $Y_0 - y_0$, where y_0 is a non ℓ-adic residue in $\mathbb{F}_q$; since we can bypass the factorization of Φ_ℓ, the cost of initialization is $O_e(\log(q))$ operations in $\mathbb{F}_q$. Besides, no descent is required: for $i \geq 0$, we have $\mathbb{K}_i = \mathbb{F}_{q^{\ell^i}}$ and $x_i = y_i$; the families of polynomials we obtain are

$$Q_i = X_i^{\ell^i} - y_0 \quad \text{and} \quad T_i = X_i^\ell - X_{i-1}. \qquad (6)$$

Lifting amounts to taking $F = \sum_{0 \leq j < \ell^{i+1}} f_j x_{i+1}^j$ and rewriting it as a bivariate polynomial in x_i, x_{i+1}, using the rule

$$x_{i+1}^j = x_i^{j \text{ div } \ell} x_{i+1}^{j \bmod \ell}.$$

Pushing does the converse operation, using the rule

$$x_i^e x_{i+1}^f = x_{i+1}^{e\ell+f}.$$

Both use only exponent arithmetic, and no operation in $\mathbb{F}_q$.

2.3 T_2-type extensions

Next, we consider the case where ℓ divides $q+1$, so that Φ_ℓ splits into quadratic factors over $\mathbb{F}_q$ (that is, $r = 2$). We also require that y_0 has norm 1 over $\mathbb{F}_q$ (see below for a discussion); we can then deduce an expression for the polynomials $Q_{i,j} \in \mathbb{F}_q(x_j)[X_i]$.

Proposition 1. *For $1 \leq j < i$, $Q_{i,j}$ satisfies*

$$Q_{i,j}(X_i) = Y^{\ell^{i-j}} + Y^{-\ell^{i-j}} - x_j \mod Y^2 - X_iY + 1. \quad (7)$$

Proof. Since $\mathrm{N}_{\mathbb{K}_0/\mathbb{F}_q}(y_0) = 1$, $\mathrm{N}_{\mathbb{K}_i/\mathbb{F}_q(x_i)}(y_i)$ is an ℓ^i-th root of unity. But ℓ does not divide $q-1$, so 1 is the only such root in $\mathbb{F}_q$, and by induction on i it also is the only root in $\mathbb{F}_q(x_i)$; hence, the minimal polynomial of y_i over $\mathbb{F}_q(x_i)$ is $Y_i^2 - x_iY_i + 1$. By composition, it follows that the minimal polynomial of y_i over $\mathbb{F}_q(x_j)$ is $Y_i^{2\ell^{i-j}} - x_jY_i^{\ell^{i-j}} + 1$. Taking a resultant to eliminate Y_i between these two polynomials gives the following relation between x_j and x_i:

$$Q_{i,j}(X_i)^2 = \mathrm{Res}_{Y_i}(Y_i^{2\ell^{i-j}} - x_jY_i^{\ell^{i-j}} + 1,\ Y_i^2 - X_iY_i + 1).$$

By direct calculation, this is equivalent to Eq. (7). □

This proposition would allow us to compute $Q_{i,j}$ in time $O(\mathsf{M}(\ell^{i-j}))$ by repeated squaring. In Section 3.1, we use arithmetic geometry to give a better algorithm, and to efficiently find a y_0 satisfying the hypotheses; we leave the algorithms for lift and push to Section 4.

2.4 The general case

Finally, we discuss the general situation, where make no assumption on the behavior of Φ_ℓ in $\mathbb{F}_q[X]$. This completes the third row of Table 1, using the bound $r \in O(\ell)$.

Because $r = [\mathbb{K}_0 : \mathbb{F}_q]$ divides $\ell - 1$, it is coprime with ℓ. Thus, Q_i remains the minimal polynomial of x_i over $\mathbb{K}_0$, and more generally $Q_{i,j}$ remains the minimal polynomial of x_i over $\mathbb{K}_j$; this will allow us to replace $\mathbb{F}_q$ by $\mathbb{K}_0$ as our base field. We will measure all costs by counting operations in $\mathbb{K}_0$, and we will deduce the cost over $\mathbb{F}_q$ by adding a factor $O(\mathsf{M}(r)\log(r))$ to account for the cost of arithmetic in $\mathbb{K}_0$.

For $i \geq 0$, since $\mathbb{K}_i = \mathbb{K}_0[Y_i]/\langle Y_i^{\ell^i} - y_0\rangle$, we represent the elements of $\mathbb{K}_i$ on the basis $\{y_i^e \mid 0 \leq e < \ell^i\}$; for instance,

x_i is written on this basis as

$$x_i = \sum_{j=0}^{r-1} y_i^{q^{\ell^i j} \bmod \ell^i} y_0^{q^{\ell^i j} \operatorname{div} \ell^i}. \tag{8}$$

Our strategy is to convert between two univariate bases of $\mathbb{K}_i$, $\{y_i^e \mid 0 \le e < \ell^i\}$ and $\{x_i^e \mid 0 \le e < \ell^i\}$. In other words, we show how to apply the isomorphism

$$\Psi_i : \mathbb{K}_i = \mathbb{K}_0[Y_i]/\langle Y_i^{\ell^i} - y_0\rangle \to \mathbb{K}_0[X_i]/\langle Q_{i,0}\rangle$$

and its inverse; we will compute the required polynomials $Q_{i,0}$ and $Q_{i,i-1}$ as a byproduct. In a second time, we will use Ψ_i to perform push and lift between the monomial basis in x_i and the bivariate basis in (x_{i-1}, x_i).

We will factor Ψ_i into elementary isomorphisms

$$\Psi_{i,j} : \mathbb{K}_j[X_i]/\langle Q_{i,j}\rangle \to \mathbb{K}_{j-1}[X_i]/\langle Q_{i,j-1}\rangle, \quad j = i, \dots, 1.$$

To start the process, with $j = i$, we let $Q_{i,i} = X_i - x_i \in \mathbb{K}_i[X_i]$, so that $\mathbb{K}_i = \mathbb{K}_i[X_i]/\langle Q_{i,i}\rangle$. Take now $j \le i$ and suppose that $Q_{i,j}$ is known. We are going to factor $\Psi_{i,j}$ further as $\Phi''_{i,j} \circ \Phi'_{i,j} \circ \Phi_{i,j}$, by introducing first the isomorphism

$$\varphi_j : \mathbb{K}_j \to \mathbb{K}_{j-1}[Y_j]/\langle Y_j^\ell - y_{j-1}\rangle.$$

The forward direction is a push from the monomial basis in y_j to the bivariate basis in (y_{j-1}, y_j) and the inverse is a lift; none of them involves any arithmetic operation (see Subsection 2.2). Then, we deduce the isomorphism

$$\Phi_{i,j} : \mathbb{K}_j[X_i]/\langle Q_{i,j}\rangle \to \mathbb{K}_{j-1}[Y_j, X_i]/\langle Y_j^\ell - y_{j-1}, Q^\star_{i,j}\rangle,$$

where $Q^\star_{i,j}$ is obtained by applying φ_j to all coefficients of $Q_{i,j}$. Since $\Phi_{i,j}$ consists in a coefficient-wise application of φ_j, applying it or its inverse costs no arithmetic operations.

Next, changing the order of Y_j and X_i, we deduce that there exists $S_{i,j}$ in $\mathbb{K}_{j-1}[X_j]$ and an isomorphism

$$\begin{aligned}\Phi'_{i,j} : \mathbb{K}_{j-1}[Y_j, X_i]/\langle Y_j^\ell - y_{j-1}, Q^\star_{i,j}\rangle \to \\ \mathbb{K}_{j-1}[X_i, Y_j]/\langle Q_{i,j-1}, Y_j - S_{i,j}\rangle,\end{aligned}$$

where $\deg(Q^\star_{i,j}, X_i) = \ell^{i-j}$ and $\deg(Q_{i,j-1}, X_i) = \ell^{i-j+1}$.

LEMMA 2. *From $Q^\star_{i,j}$, we can compute $Q_{i,j-1}$ and $S_{i,j}$ in $O(\mathsf{M}(\ell^{i+1})\log(\ell^i))$ operations in $\mathbb{K}_0$. Once this is done, we can apply $\Phi'_{i,j}$ or its inverse in $O(\mathsf{M}(\ell^{i+1}))$ operations in $\mathbb{K}_0$.*

PROOF. We obtain $Q_{i,j-1}$ and $S_{i,j}$ from the resultant and degree-1 subresultant of $Y_j^\ell - y_{j-1}$ and $Q^\star_{i,j}$ with respect to Y_j, computed over the polynomial ring $\mathbb{K}_{j-1}[X_i]$. This is done by the algorithms of [22, 20], using $O(\mathsf{M}(\ell^{i+1})\log(\ell))$ operations in $\mathbb{K}_0$ (for this analysis, and all others in this proof, we assume that we use Kronecker's substitution for multiplications). To obtain $S_{i,j}$, we invert the leading coefficient of the degree-1 subresultant modulo the resultant $Q_{i,j-1}$; this takes $O(\mathsf{M}(\ell^i)\log(\ell^i))$ operations in $\mathbb{K}_0$.

Applying $\Phi'_{i,j}$ amounts to taking a polynomial $A(Y_j, X_i)$ reduced modulo $\langle Y_j^\ell - y_{j-1}, Q^\star_{i,j}\rangle$ and reducing it modulo $\langle Q_{i,j-1}, Y_j - S_{i,j}\rangle$. This is done by computing $A(S_{i,j}, X_i)$, doing all operations modulo $Q_{i,j-1}$. Using Horner's scheme, this takes $O(\ell)$ operations $(+, \times)$ in $\mathbb{K}_{j-1}[X_i]/\langle Q_{i,j-1}\rangle$, so the complexity claim follows.

Conversely, we start from $A(X_i)$ reduced modulo $Q_{i,j-1}$; we have to reduce it modulo $\langle Y_j^\ell - y_{j-1}, Q^\star_{i,j}\rangle$. This is done using the fast Euclidean division algorithm with coefficients in $\mathbb{K}_{j-1}[Y_j]/\langle Y_j^\ell - y_{j-1}\rangle$ for $O(\mathsf{M}(\ell^{i+1}))$ operations in $\mathbb{K}_0$. □

The last isomorphism $\Phi''_{i,j}$ is trivial:

$$\Phi''_{i,j} : \mathbb{K}_{j-1}[X_i, Y_j]/\langle Q_{i,j-1}, Y_j - S_{i,j}\rangle \to \mathbb{K}_{j-1}[X_i]/\langle Q_{i,j-1}\rangle$$

forgets the variable Y_j; it requires no arithmetic operation.

Taking $j = i, \dots, 1$ allows us to compute $Q_{i,i-1}$ and $Q_{i,0}$ for $O(i^2\mathsf{M}(\ell^{i+1})\log(\ell))$ operations in $\mathbb{K}_0$. Composing the maps $\Psi_{i,j}$, we deduce further that we can apply Ψ_i or its inverse for $O(i\mathsf{M}(\ell^{i+1}))$ operations in $\mathbb{K}_0$.

We claim that we can then perform push and lift between the monomial basis in x_i and the bivariate basis in (x_{i-1}, x_i) for the same cost. Let us for instance explain how to lift.

We start from A written on the bivariate basis in (x_{i-1}, x_i); that is, A is in $\mathbb{K}_0[X_{i-1}, X_i]/\langle Q_{i-1}, T_i\rangle$. Apply Ψ_{i-1} to its coefficients in $x_i^0, \dots, x_i^{\ell-1}$, to rewrite A as an element of

$$\mathbb{K}_0[Y_{i-1}, X_i]/\langle Y_{i-1}^{\ell^{i-1}} - y_{i-2}, T_i\rangle = \mathbb{K}_{i-1}[X_i]/\langle Q_{i,i-1}\rangle.$$

Applying $\Psi_{i,i}^{-1}$ gives us the image of A in $\mathbb{K}_i$, and applying Ψ_i finally brings it to $\mathbb{K}_0[X_i]/\langle Q_i\rangle$.

3. TOWERS FROM IRREDUCIBLE FIBERS

In this section we discuss another construction of the ℓ-adic tower based on work of Couveignes and Lercier [4]. The results of this section are summarized in rows 2, 4 and 5 of Table 1. This construction is not unrelated to the ones of the previous section, and indeed we will start by showing how those of Sections 2.2 and 2.3 reduce to it.

Here is the bottom line of Couveignes' and Lercier's idea. Let G, G' be integral algebraic $\mathbb{F}_q$-groups of the same dimension and let $\phi : G' \to G$ be a surjective, separable algebraic group morphism. Let ℓ be the degree of ϕ; then, the set of points $x \in G$ with fiber G'_x of cardinality ℓ is a nonempty open subset $U \subset G$. If the induced homomorphism $G'(\mathbb{F}_q) \to G(\mathbb{F}_q)$ of groups is not surjective then there are points of $G(\mathbb{F}_q)$ with fibers lying in algebraic extensions of $\mathbb{F}_q$. Assume that we are able to choose ϕ so that we can find one of these points contained in U, with an irreducible fiber, and apply a linear projection to this fiber (e.g., onto an axis). The resulting polynomial is irreducible of degree dividing ℓ (and expectedly equal to ℓ). If we can repeat the construction with a new map $\phi' : G'' \to G'$, and so on, the sequence of extensions makes an ℓ-adic tower over $\mathbb{F}_q$.

3.1 Towers from algebraic tori

In [4], Couveignes and Lercier explain how their idea yields the tower of Section 2.2. Consider the multiplicative group $\mathbb{G}_m$: this is an algebraic group of dimension one, and $\mathbb{G}_m(\mathbb{F}_q)$ has cardinality $q - 1$. The ℓ-th power map defined by $\phi : X \mapsto X^\ell$ is a degree ℓ algebraic endomorphism of $\mathbb{G}_m$, surjective over the algebraic closure.

Suppose that ℓ divides $q - 1$, and let η be a non ℓ-adic residue in $\mathbb{F}_q$ (η plays here the same role as y_0 in Section 2). For any $i > 0$, the fiber $\phi^{-i}(\eta)$ is defined by $X^{\ell^i} - \eta$: we recover the construction of Subsection 2.2.

More generally, let $\mathbb{F}_{q^n}/\mathbb{F}_q$ be a finite extension and define its *maximal torus* as

$$T_n = \{\alpha \in \mathbb{F}_{q^n} \mid \mathrm{N}_{\mathbb{F}_{q^n}/\mathbb{F}_{q^m}}(\alpha) = 1 \text{ for any } m|n\}. \tag{9}$$

T_n is a multiplicative subgroup of $\mathbb{F}_q^*$, and, by Weil descent, an algebraic group over $\mathbb{F}_q$. It has dimension $\varphi(n)$, cardinality $\Phi_n(q)$, and is isomorphic to $\mathbb{G}_m^{\varphi(n)}$ over $\bar{\mathbb{F}}_q$ [24, 31].

We now detail how the construction of Section 2.3 can be obtained by considering the torus T_2; this will allow us to start completing the second row in Table 1.

Lemma 3. *Let $\Delta \in \mathbb{F}_q$ be a quadratic non-residue if $p \neq 2$, or such that $\mathrm{Tr}_{\mathbb{F}_q/\mathbb{F}_2}(\Delta) = 1$ otherwise. Let $\delta = \sqrt{\Delta}$ or $\delta^2 + \delta = \Delta$ accordingly. The maximal torus T_2 of $\mathbb{F}_q(\delta)/\mathbb{F}_q$ is isomorphic to the* Pell conic

$$C \ : \ \begin{cases} x^2 - \Delta y^2 = 4 & \text{if } p \neq 2, \\ x^2\Delta + xy + y^2 = 1 & \text{if } p = 2. \end{cases} \tag{10}$$

Multiplication in T_2 induces a group law on C. The neutral element is $(2,0)$ if $p \neq 2$, or $(0,1)$ if $p = 2$. The sum of two points $P = (x_1, y_1)$ and $Q = (x_2, y_2)$ is defined by

$$P \oplus Q = \begin{cases} \left(\dfrac{x_1x_2 + \Delta y_1y_2}{2}, \ \dfrac{x_1y_2 + x_2y_1}{2}\right) & \text{if } p \neq 2, \\ (x_1x_2 + x_1y_2 + x_2y_1, \ x_1x_2\Delta + y_1y_2) & \text{if } p = 2. \end{cases}$$

Proof. The isomorphism follows by Weil descent with respect to the basis $(1/2, \delta/2)$ if $p \neq 2$, or $(\delta, 1)$ if $p = 2$. Indeed, by virtue of Eq. (9), an element (x, y) of $\mathbb{F}_q(\delta)$ belongs to T_2 if and only if its norm over $\mathbb{F}_q$ is 1.

Let σ be the generator of $\mathrm{Gal}_{\mathbb{F}_q(\delta)/\mathbb{F}_q}$. For $p = 2$, clearly $\delta^\sigma = -\delta$. For $p \neq 2$, by Artin-Schreier theory, $\mathrm{Tr}_{\mathbb{F}_q(\delta)/\mathbb{F}_q}(\delta) = \mathrm{Tr}_{\mathbb{F}_q/\mathbb{F}_2}(\Delta) = 1$, hence $\delta^\sigma = 1 + \delta$. In both cases, Eq. (10) follows. The group law is obtained by direct calculation. □

Pell conics are a classic topic in number theory[18] and computer science, with applications to primality proving, factorization [17, 11] and cryptography [23].

As customary, we denote by $[n](x, y)$ the n-th scalar multiple of a point (x, y).

Lemma 4. *If $(n, p) = 1$, then $[n]$ is a separable endomorphism of C of degree n, given by the rational maps*

$$[n](x, y) = \begin{cases} \big(P_n(x), yR_n(x)\big) & \text{if } p \neq 2, \\ \big(P_n(x), yR_n(x) + R_{n-1}(x)\big) & \text{if } p = 2. \end{cases} \tag{11}$$

where P_n and R_n are defined by the initial values

$$P_0 = 2, \qquad P_1 = X,$$
$$R_0 = 0, \qquad R_1 = 1,$$

and by the same recurrence $u_{n+1} = Xu_n - u_{n-1}$.

Proof. We know that $C \cong \mathbb{G}_m$, thus $C[n] \cong \mathbb{Z}/n\mathbb{Z}$ and $[n]$ is separable of degree n. Eq. (11) is shown by induction using Eq. (10) and the group law. □

Theorem 5. *Let $\eta \in \mathbb{F}_q(\delta)$ be a non ℓ-adic residue in T_2, and let $P = (\alpha, \beta)$ be its image in $C/\mathbb{F}_q$. For any $i > 0$, the polynomials $P_{\ell^i} - \alpha$ are irreducible. Their roots are the abscissas of the images in $C/\mathbb{F}_{q^{\ell^i}}$ of the ℓ^i-th roots of η.*

Proof. By [15, Th. VI.9.1], the polynomial $X^{\ell^i} - \eta$ is irreducible. Its roots correspond to the fiber $[\ell^i]^{-1}(P)$, and the Galois group of $\mathbb{F}_{q^{\ell^i}}/\mathbb{F}_q$ acts transitively on them.

Two points of C have the same abscissa if and only if they are opposite. But $\eta \neq \eta^{-1}$, hence all the points in $[\ell^i]^{-1}(P)$ have distinct abscissa. By Lemma 4, $P_{\ell^i} - \alpha$ vanishes precisely on those abscissas and is thus irreducible. □

We can now apply our results to the computation of the polynomials Q_i and T_i of Section 2.3.

Corollary 6. *The polynomials $Q_{i,j}$ of Prop. 1 satisfy*

$$Q_{i,j}(X_i) = P_{\ell^{i-j}}(X_i) - x_j.$$

Proof. We have already shown that $\mathrm{N}_{\mathbb{K}_j/\mathbb{F}_q(x_j)}(y_j) = 1$ for any j, thus y_j is a non ℓ-adic residue in $T_2/\mathbb{F}_q(x_j)$. Independently of the characteristic and of the element $\Delta \in \mathbb{F}_q(x_j)$ chosen, the abscissa of the image of y_j in $C/\mathbb{F}_q(x_j)$ is $\mathrm{Tr}_{\mathbb{K}_j/\mathbb{F}_q(x_j)}\, y_j = x_j$. The statement follows from the previous theorem. □

Corollary 7. *The polynomials $Q_{i,j}$ can be computed using $O(\ell^{i-j})$ operations.*

Proof. From the previous corollary, it is enough to compute P_n using $O(n)$ operations. We write $P_n = \sum_i c_{n,i}X^{n-i}$, from Lemma 4 we deduce that

$$c_{n,i} = c_{n-1,i} - c_{n-2,i-2}. \tag{12}$$

By induction, it is immediate that $c_{n,i} = 0$ for i odd, and that signs alternate for i even, so we remove the odd coefficients and take absolute values. The new coefficients $b_{n,k} = |c_{n+k,2k}|$ satisfy the relation

$$b_{n,k} = b_{n-1,k} + b_{n-1,k-1},$$

which is the same as Pascal's relation; we actually obtain the $(1, 2)$-Pascal triangle, also called Lucas' triangle [1]. In the same way, we can prove that the even coefficients of R_n are the entries of Pascal's triangle with alternating signs.

As is well-known, the coefficients of Lucas' triangle are related to those of Pascal's by

$$b_{n,k} = \binom{n}{k} + \binom{n-1}{k-1} = \frac{n+k}{n}\binom{n}{k}. \tag{13}$$

Using Eq. (13) and the sign alternation property, we get

$$\frac{c_{n,2k+2}}{c_{n,2k}} = -\frac{(n-2k)(n-2k-1)}{(n-k-1)(k+1)}.$$

The last equation gives the formula to compute all the coefficients of P_n using $O(n)$ operations in $\mathbb{F}_p$. Indeed, since we know the $c_{n,2k}$'s are the image mod p of integers, we compute them using multiplications and divisions in $\mathbb{Q}_p$ with relative precision 1. □

We are left with the problem of finding the non ℓ-adic residue η to initialize the tower. As before, this will be done by random sampling and testing.

Lemma 8. *Let $P = (\alpha, \beta)$ be a point on C. For any n, there is a formula to compute the abscissa of $[\pm n]P$, using $O(\log n)$ operations in $\mathbb{F}_q$, and not involving β.*

Proof. Observe that if $n = 2$, the abscissa of $[\pm 2]P$ is $\alpha^2 - 2$ (for any p). Let $P' = (\alpha', \beta')$, and let γ be the abscissa of $P \ominus P'$. By direct computation we find that the abscissa of $P \oplus P'$ is $\alpha\alpha' - \gamma$ (for any p); this formula is called a *differential addition.* Thus, $O(1)$ operations are needed for a doubling or a differential addition. To compute the abscissa of $[\pm n]P$, we use the ladder algorithm of [21], requiring $O(\log n)$ doublings and differential additions. □

Proposition 9. *The abscissa of a point $P \in C/\mathbb{F}_q$ satisfying the conditions of Theorem 5 can be found using $O_e(\log q)$ operations in $\mathbb{F}_q$.*

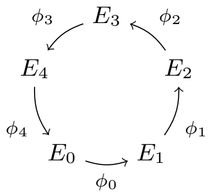

Figure 2: The isogeny cycle of E_0.

Proof. We randomly select $\alpha \in \mathbb{F}_q$ and test that it belongs to C. If $p \neq 2$, this amounts to testing that $\alpha^2 - 4$ is a quadratic non-residue in $\mathbb{F}_q$, a task that can be accomplished with $O(\log q)$ operations. If $p = 2$, by Artin-Schreier theory this is equivalent to $\mathrm{Tr}_{\mathbb{F}_q/\mathbb{F}_2}(1/\alpha^2) = 1$, which can be tested in $O(\log q)$ operations in $\mathbb{F}_q$.

Then we check that P is a non ℓ-adic residue by verifying that $[(q+1)/\ell]P$ is not the group identity. By Lemma 8, this computation requires $O(\log q)$ operations. About half of the points of $\mathbb{F}_q$ are quadratic non-residues, and about $1-1/\ell$ of them are the abscissas of points with the required order, thus we expect to find the required element after $O_e(1)$ trials. □

It is natural to ask whether a similar construction could be applied to any ℓ. If r is the order of q modulo ℓ, the natural object to look at is T_r, but here we are faced with two problems. First, multiplication by ℓ is now a degree $\ell^{\varphi(r)}$ map, thus its fibers have too many points; instead, isogenies of degree ℓ should be considered. Second, it is an open question whether T_r can be parameterized using $\varphi(r)$ coordinates; but even assuming it can be, we are still faced with the computation of a univariate annihilating polynomial for a set embedded in a $\varphi(r)$-dimensional space, a problem not known to be feasible in quasi-linear time. Studying this generalization is another natural follow-up to the present work.

3.2 Towers from elliptic curves

Since it seems hard to deal with higher dimensional algebraic tori, it is interesting to look at other algebraic groups. Being one-dimensional, elliptic curves are good candidates. In this section, we quickly review Couveignes' and Lercier's construction, referring to [4] for details, and point out the modifications needed in order to build towers (as opposed to constructing irreducible polynomials).

Let ℓ be a prime different from p and not dividing $q-1$. Let E_0 be an elliptic curve whose cardinality is a multiple of ℓ. By Hasse's bound, this is only possible if $\ell \leq q + 2\sqrt{q} + 1$. An *isogeny* is an algebraic group morphism between two elliptic curves that is surjective in the algebraic closure. It is said to be rational over $\mathbb{F}_q$ if it is invariant under the q-th power map; such an isogeny exists if and only if the curves have the same number of points over $\mathbb{F}_q$. An isogeny of degree n is separable if and only if n is prime to p, in which case its kernel contains exactly n points. Because of the assumptions on ℓ, there exists an $e \geq 1$ such that, for any curve E isogenous to E_0, the $\mathbb{F}_q$-rational part of $E[\ell]$ is cyclic of order ℓ^e.

Suppose for simplicity, that $p \neq 2, 3$ and let E_0 be expressed as the locus

$$E_0 \;:\; y^2 = x^3 + ax + b, \quad \text{with } a, b \in \mathbb{F}_q, \tag{14}$$

plus one point at infinity. We denote by H_0 the unique subgroup of $E_0/\mathbb{F}_q$ of order ℓ, and by ϕ_0 the unique isogeny whose kernel is H_0; we then label E_1 the image curve of ϕ_0. We go on denoting by H_i the unique subgroup of $E_i/\mathbb{F}_q$ of order ℓ, and by $\phi_i : E_i \to E_{i+1}$ the unique isogeny with kernel H_i. The construction is depicted in Figure 2.

Lemma 10. *Let $E_0, E_1, \dots$ be defined as above, there exists $n \in O(\sqrt{q}\log(q))$ such that E_n is isomorphic to E_0.*

Proof. It is shown in [4, § 4] that the isogenies ϕ_i are *horizontal* in the sense of [14], hence they necessarily form a cycle. Let t be the trace of E_0, the length of the cycle is bounded by the class number of $\mathbb{Q}[X]/(X^2 - tX - q)$, thus by Minkowski's bound it is in $O(\sqrt{q}\log(q))$. □

In what follows, the index i is to be understood modulo the length of the cycle. This is a slight abuse, because E_n is isomorphic but not equal to E_0, but it does not hide any theoretical or computational difficulty.

Under the former assumptions, it is proved in [4, § 4] that if P is a point of E_i of order divisible by ℓ^e, if $\psi = \phi_{i-1} \circ \phi_{i-2} \circ \cdots \circ \phi_j$, then the fiber $\psi^{-1}(P)$ is irreducible and has cardinality ℓ^{i-j}. Knowing E_i, Vélu's formulas [29] allow us to express the isogenies ϕ_i as rational fractions

$$\begin{aligned} \phi_i &: E_i \to E_{i+1}, \\ (x, y) &\mapsto \left(\frac{f_i(x)}{g_i(x)}, y\left(\frac{f_i(x)}{g_i(x)}\right)'\right), \end{aligned} \tag{15}$$

where g_i is the square polynomial of degree $\ell-1$ vanishing on the abscissas of the affine points of H_i, and f_i is a polynomial of degree ℓ.

There is a subtle difference between our setting and Couveignes' and Lercier's. The goal of [4] is to compute an extension of degree ℓ^i of $\mathbb{F}_q$ for a fixed i: this can be done by going forward i times, then taking the fiber of a point of E_i by the isogenies $\phi_{i-1}, \dots, \phi_0$. In our case, we are interested in building extensions of degree ℓ^i *incrementally*, i.e. without any *a priori* bound on i. Thus, we have to walk *backwards* in the isogeny cycle: if $\eta \in \mathbb{F}_q$ is the abscissa of a point of E_0 of order $\ell^e \neq 2$, we will use the following polynomials to define the ℓ-adic tower:

$$\begin{aligned} T_1 &= f_{-1}(X_1) - \eta g_{-1}(X_1), \\ T_i &= f_{-i}(X_i) - X_{i-1} g_{-i}(X_i). \end{aligned}$$

The following theorem gives the time for building the tower; lift and push are detailed in the next section.

Theorem 11. *Suppose $4\ell \leq q^{1/4}$, and under the above assumption. Initializing the ℓ-adic tower requires $O_{\tilde{e}}(\ell\log^5(q) + \ell^3)$ bit operations; and building the i-th level requires $O_e(\ell^2 + \mathsf{M}(\ell)\log(\ell q) + \mathsf{M}(\ell^i)\log(\ell^i))$ operations in $\mathbb{F}_q$.*

Proof. For the initialization, [4, § 4.3] shows that if $4\ell \leq q^{1/4}$, a curve E_0 with the required number of points can be found in $O_{\tilde{e}}(\ell\log^5(q))$ bit operations. We also need to compute the ℓth modular polynomial Φ_ℓ mod p; for this, we compute it over $\mathbb{Z}$ with $\tilde{O}(\ell^3)$ bit operations [8], then reduce it modulo p.

To build the i-th level, we first need to find the equation of E_{-i}. For this, we evaluate Φ_ℓ at $j(E_{-i+1})$, using $O(\ell^2)$ operations. The resulting polynomial has two roots in $\mathbb{F}_q$, namely $j(E_{-i})$ and $j(E_{-i+2})$. We factor it using $O_e(\mathsf{M}(\ell)\log(\ell q))$ operations [30, Ch 14]. Once E_{-i} is known, we find an ℓ-torsion point using $O_e(\log q)$ operations, and apply Vélu's

Algorithm 1 Compose

```
Input: $P \in \mathbb{F}_q[X,Y]$, $f, g \in \mathbb{F}_q[Y]$, $n \in \mathbb{N}$
 1: if $n = 1$ then
 2:     return $P$
 3: else
 4:     $m \leftarrow \lceil n/2 \rceil$
 5:     Let $P_0, P_1$ be such that $P = P_0 + X^m P_1$
 6:     $Q_0 \leftarrow$ Compose$(P_0, f, g, m)$
 7:     $Q_1 \leftarrow$ Compose$(P_1, f, g, n-m)$
 8:     $Q \leftarrow Q_0 g^{n-m} + Q_1 f^m$
 9:     return $Q$
10: end if
```

formulas to compute ϕ_{-i}. We deduce the polynomial T_i, and Q_i is obtained using $O(\mathsf{M}(\ell^i)\log(\ell^i))$ operations using Algorithm 1 given in the next section. □

Remark 1. Instead of computing the cycle step by step, we could compute it entirely during the initialization phase, by using Vélu's formulas alone to compute $E_1, E_2, \dots$ until we hit E_0 again. By doing so, we avoid using the modular polynomial Φ_ℓ at each new level. By Lemma 10, this requires $O_e(\ell\sqrt{q}\log(q))$ operations. This is not asymptotically good in q, but for practical values of q and ℓ the cycle is often small and this approach works well. This is accounted for in the last row of Table 1.

4. LIFTING AND PUSHING

The previous constructions of ℓ-adic towers based on irreducible fibers share a common structure that allows us to treat lifting and pushing in a unified way. Renaming the variables (X_{i-1}, X_i) as (X, Y), the polynomials (Q_{i-1}, Q_i, T_i) as (R, S, T), the extension at level i is described as

$$\mathbb{F}_q[Y]/\langle S(Y)\rangle \quad \text{and} \quad \mathbb{F}_q[X,Y]/\langle R(X), T(X,Y)\rangle,$$

with R of degree ℓ^{i-1}, S of degree ℓ^i, and where $T(X,Y)$ has the form $f(Y) - Xg(Y)$, with $\deg(f) = \ell$, $\deg(g) < \ell$ and $\gcd(f,g) = 1$; possibly, $g = 1$. In all this section, f, g and their degree ℓ are fixed.

Lift is the conversion from the bivariate basis associated to the right-hand side to the univariate basis associated to the left-hand side; push is the inverse. Using the special shape of the polynomial T, they reduce to composition and decomposition of rational functions, as we show next. These results fill in all missing entries in the lift / push column of Table 1.

4.1 Lifting

Let P be in $\mathbb{F}_q[X,Y]$ and n be in $\mathbb{N}$, with $\deg(P,X) < n$. We define $P[f,g,n]$ as

$$P[f,g,n] = g^{n-1} P\left(\frac{f}{g}, Y\right) \in \mathbb{F}_q[X,Y].$$

If $P = \sum_{i=0}^{n-1} p_i(Y) X^i$, then $P[f,g,n] = \sum_{i=0}^{n-1} p_i f^i g^{n-1-i}$. We first give an algorithm to compute this expression, then show how to relate it to lifting; when $g = 1$, Algorithm 1 reduces to a well known algorithm for polynomial composition [30, Ex. 9.20].

Theorem 12. *On input P, f, g, n, with $\deg(P,X) < n$ and $\deg(P,Y) < \ell$, Algorithm 1 computes $Q = P[f,g,n]$ using $O(\mathsf{M}(\ell n)\log(n))$ operations in $\mathbb{F}_q$.*

Algorithm 2 Decompose

```
Input: $Q, f, g, h \in \mathbb{F}_q[Y]$, $n \in \mathbb{N}$
 1: if $n = 1$ then
 2:     return $Q$
 3: else
 4:     $m \leftarrow \lceil n/2 \rceil$
 5:     $u \leftarrow 1/g^{n-m} \bmod f^m$
 6:     $Q_0 \leftarrow Qu \bmod f^m$
 7:     $Q_1 \leftarrow (Q - Q_0 g^{n-m})$ div $f^m$
 8:     $P_0 \leftarrow$ Decompose$(Q_0, f, g, h, m)$
 9:     $P_1 \leftarrow$ Decompose$(Q_1, f, g, h, n-m)$
10:     return $P_0 + X^m P_1$
11: end if
```

Proof. If $n = 1$, the theorem is obvious. Suppose $n > 1$, then P_0 and P_1 have degrees less than m and $n - m$ respectively. By induction hypothesis,

$$Q_0 = P_0[f,g,m] = \sum_{i=0}^{m-1} p_i f^i g^{m-1-i},$$

$$Q_1 = P_1[f,g,n-m] = \sum_{i=0}^{n-m-1} p_{i+m} f^i g^{n-m-1-i}.$$

Hence,

$$Q = \sum_{i=0}^{m-1} p_i f^i g^{n-1-i} + \sum_{i=0}^{n-m-1} p_{i+m} f^{i+m} g^{n-m-1-i} = P[f,g,n].$$

The only step that requires a computation is Step 8, costing $O(\mathsf{M}(\ell n))$ operations in $\mathbb{F}_q$. The recursion has depth $\log(n)$, hence the overall complexity is $O(\mathsf{M}(\ell n)\log(n))$. □

Corollary 13. *At level i, one can perform the lift operation using $O(\mathsf{M}(\ell^i)\log(\ell^i))$ operations in $\mathbb{F}_q$.*

Proof. We start from an element α written on the bivariate basis, that is, represented as $A(X,Y)$ with $\deg(A,X) < n = \ell^{i-1}$ and $\deg(A,Y) < \ell$ (note that $\ell n = \ell^i$). We compute the univariate polynomials $A^\star = A[f,g,n]$ and $\gamma = g^{n-1}$ using $O(\mathsf{M}(\ell^i)\log(\ell^i))$ operations in $\mathbb{F}_q$; then the lift of α is $A^\star/\gamma$ modulo S. The inverse of γ is computed using $O(\mathsf{M}(\ell n)\log(\ell n))$ operations, and the multiplication adds an extra $O(\mathsf{M}(\ell n))$. □

4.2 Pushing

We first deal with the inverse of the question dealt with in Theorem 12: starting from $Q \in \mathbb{F}_q[Y]$, reconstruct $P \in \mathbb{F}_q[X,Y]$ such that $Q = P[f,g,n]$. When $g = 1$, Algorithm 2 reduces to Algorithm 9.14 of [30].

Theorem 14. *On input Q, f, g, h, n, with $\deg(Q) < \ell n$ and $h = 1/g \bmod f$, Algorithm 2 computes a polynomial $P \in \mathbb{F}_q[X,Y]$ such that $\deg(P,X) < n$, $\deg(P,Y) < \ell$ and $Q = P[f,g,n]$ using $O(\mathsf{M}(\ell n)\log(n))$ operations in $\mathbb{F}_q$.*

Proof. We prove the theorem by induction. If $n = 1$, the statement is obvious, so let $n > 1$. The polynomials Q_0 and Q_1 verify $Q = Q_0 g^{n-m} + Q_1 f^m$. By construction, Q_0 has degree less than ℓm. Since $\deg(g) < \ell$, this implies that $Q_0 g^{n-m}$ has degree less than ℓn; thus, Q_1 has degree less than $\ell(n-m)$. By induction, P_0 and P_1 have degree less than m, resp. $n-m$, in X, and less than ℓ in Y, and

$$Q_0 = P_0[f,g,m] = \sum_{i=0}^{m-1} p_{0,i} f^i g^{m-1-i},$$

$$Q_1 = P_1[f,g,n-m] = \sum_{i=0}^{n-m-1} p_{1,i} f^i g^{n-m-1-i}.$$

Hence, $P = P_0 + X^m P_1$ has degree less than n in X and less than ℓ in Y, and the following proves correctness:

$$\begin{aligned} P[f,g,n] &= \sum_{i=0}^{m-1} p_{0,i} f^i g^{n-1-i} + \sum_{i=m}^{n-1} p_{1,i-m} f^i g^{n-1-i} \\ &= P_0[f,g,m] g^{n-m} + P_1[f,g,n-m] f^m = Q. \end{aligned}$$

At Step 5, we do as follows: starting from $h = 1/g \bmod f$, we deduce $1/g^{n-m} \bmod f$ in time $O(\mathsf{M}(\ell)\log(n))$ by binary powering mod f. We also compute g^{n-m} in time $O(\mathsf{M}(\ell n))$ by binary powering, and we use Newton iteration (starting from $1/g^{n-m} \bmod f$) to deduce $1/g^{n-m} \bmod f^m$ in time $O(\mathsf{M}(\ell n))$. All other steps cost $O(\mathsf{M}(\ell n))$; the recursion has depth $\log(n)$, so the total cost is $O(\mathsf{M}(\ell n)\log(n))$. □

Corollary 15. *At level i, one can perform the push operation using $O(\mathsf{M}(\ell^i)\log(\ell^i))$ operations in $\mathbb{F}_q$.*

Proof. Given α represented by a univariate polynomial $A(Y)$ of degree less than ℓn, with $n = \ell^{i-1}$. We compute g^{n-1} and $A^\star = g^{n-1} A \bmod S$ using $O(\mathsf{M}(\ell^i))$ operations. Then, we compute $h = 1/g \bmod f$ in time $O(\mathsf{M}(\ell)\log(\ell))$ and apply Algorithm 2 to $A^\star$, f, g, h and n. The result is a bivariate polynomial B, representing α on the bivariate basis. The dominant phase is Algorithm 2, costing $O(\mathsf{M}(\ell^i)\log(\ell^i))$ operations in $\mathbb{F}_q$. □

5. IMPLEMENTATION

To demonstrate the interest of our constructions, we made a very basic implementation of the towers of Sections 3.1 and 3.2 in Sage [28]. It relies on Sage's default implementation of quotient rings of $\mathbb{F}_p[X]$, which itself uses NTL [27] for $p = 2$ and FLINT [12] for other primes. Towers based on elliptic curves are constructed using the algorithm described in Remark 1. The source code is available on De Feo's web page.

We compare our implementation against three ways of constructing ℓ-adic towers in Magma:

- We construct the levels from bottom to top using the default finite field constructor `GF()`. For the parameters we were able to test, Magma uses tables of precomputed Conway polynomials and automatically computes embeddings on creation.[1]

- We construct the highest level of the tower first, then all the lower levels using the `sub<>` constructor.

- We construct the levels from bottom to top using random dense polynomials, then we call the `Embed()` function. We do not account for the time spent finding the irreducible polynomials.

We ran tests on an Intel Xeon E5620 clocked at 2.4 GHz, using Sage 5.5 and Magma 2.18.12. The time required for the creation of 3-adic towers of increasing height is summarized in Figure 3; the timings of our algorithms are labeled T_2 and Elliptic. Computations that took more than 4GB RAM were interrupted.

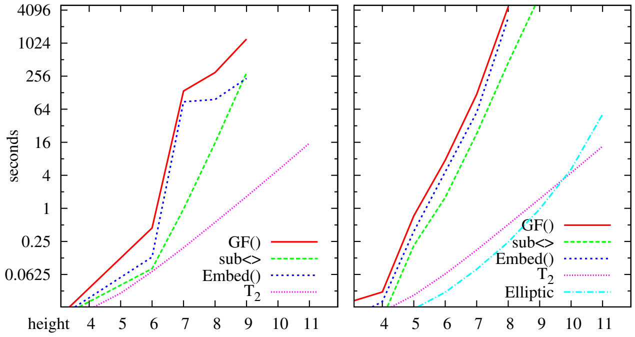

Figure 3: Times for building 3-adic towers on top of $\mathbb{F}_2$ (left) and $\mathbb{F}_5$ (right), in Magma (first three lines) and using our code.

Despite its simplicity, our code consistently outperforms Magma on creation time. On the other hand, lift and push operations take essentially no time in Magma, while in all the tests of Figure 3 we measured a running time almost perfectly linear for one push followed by one lift, taking approximately $70\mu s$ per coefficient (this is in the order of a second around level 10). Nevertheless, the large gain in creation time makes the difference in lift and push tiny, and we are convinced that an optimized C implementation of the algorithms of Section 4 would match Magma's performances.

Aknowledgements. De Feo would like to thank Antoine Joux and Jérôme Plût for fruitful discussions. Schost acknowledges support from NSERC and the CRC program.

6. REFERENCES

[1] A. T. Benjamin. The Lucas triangle recounted. In *Congressus Numerantum, Proceedings of the Twelfth Conference on Fibonacci Numbers and their Applications*, volume 200, pages 169–177, 2010.

[2] W. Bosma, J. Cannon, and C. Playoust. The MAGMA algebra system I: the user language. *Journal of Symbolic Computation*, 24(3-4):235–265, 1997.

[3] W. Bosma, J. Cannon, and A. Steel. Lattices of compatibly embedded finite fields. *Journal of Symbolic Computation*, 24(3-4):351–369, 1997.

[4] J.-M. Couveignes and R. Lercier. Fast construction of irreducible polynomials over finite fields. *To appear in the Israel Journal of Mathematics*, July 2011.

[5] L. De Feo. Fast algorithms for computing isogenies between ordinary elliptic curves in small characteristic. *Journal of Number Theory*, 131(5):873–893, May 2011.

[6] L. De Feo and E. Schost. Fast arithmetics in Artin-Schreier towers over finite fields. *Journal of Symbolic Computation*, 47(7):771–792, July 2012.

[7] J. Doliskani and É. Schost. A note on computations in degree 2^k-extensions of finite fields, 2012. Manuscript.

[8] A. Enge. Computing modular polynomials in quasi-linear time. *Mathematics of Computation*, 78(267):1809–1824, 2009.

[9] P. Gaudry and E. Schost. Point-counting in genus 2 over prime fields. *Journal of Symbolic Computation*, 47(4):368–400, 2012.

[10] S. Gurak. Minimal polynomials for gauss periods with f=2. *Acta Arithmetica*, 121(3):233, 2006.

[11] S. A. Hambleton. Generalized Lucas-Lehmer tests using Pell conics. *Proceedings of the American Mathematical Society*, 140:2653–2661, 2012.

[1] See `http://magma.maths.usyd.edu.au/magma/releasenotes/2/14`

[12] W. Hart. Fast library for number theory: an introduction. *International Conference on Mathematical Software–ICMS 2010*, pages 88–91, 2010.

[13] K. S. Kedlaya and C. Umans. Fast polynomial factorization and modular composition. *SIAM J. Computing*, 40(6):1767–1802, 2011.

[14] D. Kohel. *Endomorphism rings of elliptic curves over finite fields*. PhD thesis, University of California at Berkley, 1996.

[15] S. Lang. *Algebra*. Springer, 3rd edition, Jan. 2002.

[16] R. Lebreton and É. Schost. Algorithms for the universal decomposition algebra. In *ISSAC'12*, pages 234–241. ACM, 2012.

[17] F. Lemmermeyer. Conics - a Poor Man's Elliptic Curves, Nov. 2003.

[18] H. W. Lenstra. Solving the Pell equation. *Notices of the AMS*, 49(2):182–192, 2002.

[19] H. W. Lenstra and B. De Smit. Standard models for finite fields: the definition, 2008.

[20] T. Lickteig and M. Roy. Sylvester–habicht sequences and fast cauchy index computation. *Journal of Symbolic Computation*, 31(3):315 – 341, 2001.

[21] P. L. Montgomery. Speeding the pollard and elliptic curve methods of factorization. *Mathematics of Computation*, 48(177), 1987.

[22] D. Reischert. Asymptotically fast computation of subresultants. In *ISSAC*, pages 233–240. ACM, 1997.

[23] K. Rubin and A. Silverberg. Torus-Based cryptography. In D. Boneh, editor, *Advances in Cryptology - CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 349–365, Berlin, Heidelberg, 2003. Springer Berlin / Heidelberg.

[24] K. Rubin and A. Silverberg. Algebraic tori in cryptography. In *In High Primes and Misdemeanours: Lectures in Honour of the 60th birthday of Hugh Cowie Williams*, volume 41 of *Fields Institute Communications*. American Mathematical Society, 2004.

[25] V. Shoup. New algorithms for finding irreducible polynomials over finite fields. *Math. Comp.*, 54:435–447, 1990.

[26] V. Shoup. Fast construction of irreducible polynomials over finite fields. *Journal of Symbolic Computation*, 17(5):371–391, 1994.

[27] V. Shoup. NTL: A library for doing number theory. `http://www.shoup.net/ntl`, 2003.

[28] W. A. Stein and Others. *Sage Mathematics Software (Version 5.5)*. The Sage Development Team, 2013.

[29] J. Vélu. Isogénies entre courbes elliptiques. *Comptes Rendus de l'Académie des Sciences de Paris*, 273:238–241, 1971.

[30] J. von zur Gathen and J. Gerhard. *Modern computer algebra*. Cambridge University Press, New York, NY, USA, 1999.

[31] V. E. Voskresenskiĭ. *Algebraic groups and their birational invariants*, volume 179. American Mathematical Society, 1998.